

GPRS 通讯程序编码练习

1 练习目的

- 了解 GPRS 通讯原理;
- 掌握 AT 指令集中常用的 AT 指令的意义;
- 掌握 GPRS 发送短信、接打电话等程序的实现方法。

2 练习内容

- 调试 gprs 程序示例, 了解 gprs 初始化, 以及拨打电话的实现方法。
- 编写 gprs 通讯程序, 实现拨打、接听、挂断电话, 以及发送端信息。

3 练习原理

3.1 GPRS 模块 SIM100-E 电器特性

ARM 嵌入式开发平台的 GPRS 扩展板采用的 GPRS 模块型号为 SIM100-E, 是 SIMCOM 公司推出的 GSM/GPRS 双频模块, 主要为语音传输、短消息和数据业务提供无线接口。

SIM100-E 集成了完整的射频电路和 GSM 的基带处理器, 适合于开发一些 GSM/GPRS 的无线应用产品, 如移动电话、PCMCIA 无线 MODEM 卡、无线 POS 机、无线抄表系统以及无线数据传输业务, 应用范围十分广泛。

SIM100-E 模块为用户提供了功能完备的系统接口。60Pin 系统连接器是 SIM100-E 模块与应用系统的连接接口, 主要提供外部电源、RS-232 串口、SIM 卡接口和音频接口。SIM100-E 模块使用锂电池、镍氢电池或者其他外部直流电源供电, 电源电压范围为: 3.3V~4.6 V, 电源应该具有至少 2A 的峰值电流输出能力。

SIM100-E 提供标准的 RS-232 串行接口, 用户可以通过串行口使用 AT 命令完成对模块的操作。串行口支持以下通信速率:

300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200。

当模块上电启动并报出 RDY 后, 用户才可以和模块进行通信, 用户可以首先使用模块默认速率 115200 与模块通信, 并可通过 AT +IPR=<rate> 命令自由切换至其它通信速率。在应用设计中, 当 MCU 需要通过串口与模块进行通讯时, 可以只用三个引脚: TXD, RXD 和 GND。其他引脚悬空, 建议 RTS 和 DTR 置低。本扩展板上采用 MAX3232 芯片完成 GPRS 模块的 TTL 电平到 RS232 电平的转换, 以能和 ARM 开发平台的 RS232 串口连接。

SIM100-E 模块提供了完整的音频接口, 应用设计只需增加少量外围辅助元器件, 主要是为 MIC 提供工作电压和射频旁路。音频分为主通道和辅助通道两部分。可以通过 AT+CHFA 命令切换主副音频通道。音频设计应

该尽量远离模块的射频部分，以降低射频对音频的干扰。本扩展板硬件支持两个语音通道，主通道可以插普通电话机的话柄，辅助通道可以插带 MIC 的耳慢。

当选择为主通道时，有电话呼入时板载蜂鸣器将发出铃声以提示来电。但选择辅助通道时来电提示音乐只能在耳机中听到。蜂鸣器是由 GPRS 模块的 BUZZER 引脚加驱动电路控制的。

GPRS 模块的射频部分支持 GSM900/DCS1800 双频，为了尽量减少射频信号在射频连接线上的损耗，必须谨慎选择射频连线。应采用 GSM900/DCS1800 双频段天线，天线应满足阻抗 50 欧姆和收发驻波比小于 2 的要求。为了避免过大的射频功率导致 GPRS 模块的损坏，在模块上电前请确保天线已正确连接。

模块支持外部 SIM 卡，可以直接与 3.0V SIM 卡或者 1.8V SIM 卡连接。模块自动监测和适应 SIM 卡类型。对用户来说，GPRS 模块实现的就是一个移动电话的基本功能，该模块正常的工作是需要电信网络支持的，需要配备一个可用的 SIM 卡，在网络服务计费方面和普通手机类似。

3.2 AT 命令集简要说明

GPRS 模块和应用系统是通过串口连接的，控制系统可以发给 GPRS 模块 AT 命令的字符串来控制其行为。GPRS 模块具有一套标准的 AT 命令集。

1、一般 AT 命令

AT 命令	描述
AT+CGMI	返回生产厂商标识。
AT+CGMM	返回产品型号标识。
AT+CGMR	返回软件版本标识。
ATI	发行的产品信息。
ATE =<value>	决定是否回显输入的命令。value=0 表示关闭回显，1 打开回显。
AT+CGSN	返回产品序列号标识。
AT+CLVL?	读取受话器音量级别。
AT+CLVL=<level>	设置受话器音量级别，level 在 0~100 之间，数值越小则音量越轻。
AT+CHFA=<state>	切换音频通道。State=0 为主音频通道，1 为辅助音频通道。
AT+CMIC=<ch>,<gain>	改变 MIC 增益，ch=0 为主 MIC，1 为辅助 MIC；gain 在 0~15 之间。

2、网络服务相关命令

AT+CNUM=?	读取本机号码。
AT+COPN	读取网络运营商名称。
AT+CSQ	信号强度指示，返回接收信号强度指示值和信道误码率。

3、呼叫控制命令

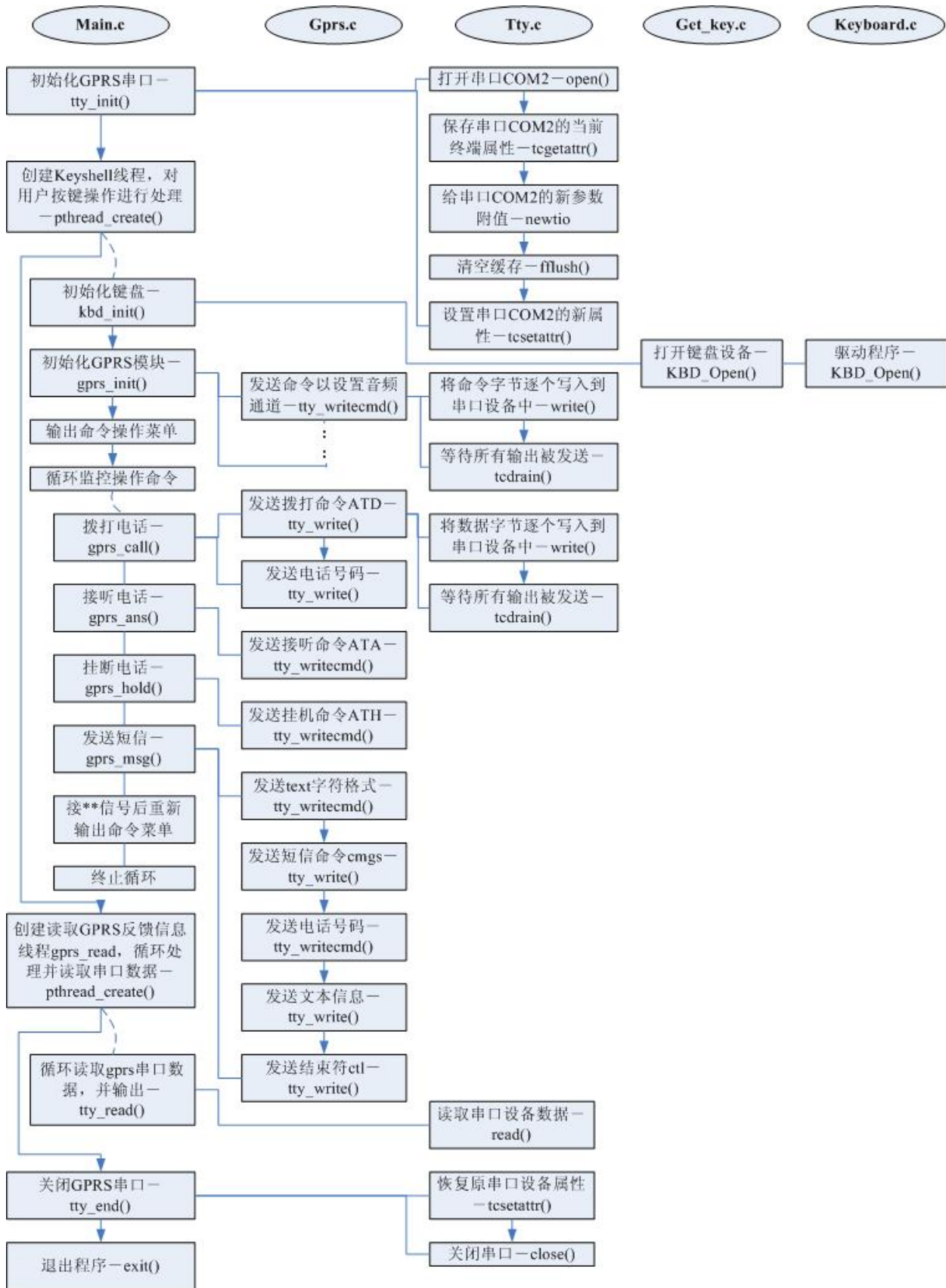
ATDxxxxxxxx;	拨打电话号码 xxxxxxxx，注意最后要加分号，中间无空格。
ATA	接听电话。
ATH	拒接电话或挂断电话。
AT+VTS=<dtmfstr>	在语音通话中发送 DTMF 音，dtmfstr 举例：“4， 5 ， 6”为 456 三字符。

4、短消息命令

AT+CMGC	发出一条短消息命令。
AT+CMGD=<index>	删除短消息。index 为所要删除短信的记录号。
AT+CMGF=<mode>	选择短消息格式。mode=0 为 PDU 模式，1 为文本模式。建议文本模式。
AT+CMGL=<stat>	列出当前短消息存储器中的短信。stat 参数空白或 0/"REC UNREAD" 为未读，1/"REC READ" 为已读，2/"STOU NSENT" 为待发，3/"STO SENT" 为已发，4/"ALL" 为全部的。
AT+CMGR=<index>	读取短消息。index 为所要读取短信的记录号。
AT+CMGS=xxxxxxxx 'CR' Text 'CTRL+Z'	发送短消息。xxxxxxxx 为对方手机号码，回车后接着输入短信内容，然后按 CTRL+Z 发送短信。CTRL+Z 的 ASCII 码是 26。
AT+CNMI	显示新收到的短消息。
AT+CSCA?	读取短消息中心地址。
AT+CSMP	设置短消息文本模式参数。

3.3 程序结构图

程序结构如下图所示：



3.4 练习步骤

3.4.1 补充拨打电话模块的代码，并编译、运行。

- 1) 阅读 main.c 文件，分析主函数 main() 的代码流程；
- 2) 阅读线程处理函数 keyshell() 中的实现代码并分析其流程；
- 3) 进入 gprs_init() 函数的定义文件 gprs.c 中，补充 gprs 初始化代码，如下：

```
void gprs_init()
{
    tty_writecmd("at", strlen("at"));
    tty_writecmd("at+chfa=1", strlen("at+chfa=1"));
    //设置音频通道：1，副音频通道
    tty_writecmd("at+chfa=1", strlen("at+chfa=1"));
    //设置输出音频信号增益：100，最大音量
    tty_writecmd("at+clvl=100", strlen("at+clvl=100"));
    //设置通道1的话筒增益：10，最大增益
    tty_writecmd("at+cmic=1,10", strlen("at+cmic=1,10"));
}
```

- 4) 分析函数 tty_writecmd() 的作用；
- 5) 在 while 循环过程中编写代码，实现 gprs 的命令菜单的输出；具体如下：

```
printf("\n< gprs control shell >");
printf("\n [1]   give a call");
printf("\n [2]   respond a call");
printf("\n [3]   hold a call");
printf("\n [4]   send a msg");
printf("\n [*]   help menu");
printf("\n [--]  exit");
```

- 6) 分析函数 get_line() 的实现代码及作用；
- 7) 分析如何通过 if 判断实现菜单各项功能（拨打、接收、发短信）的？

利用循环采集从键盘输入的信息，并执行相应的功能函数。以按键按下“1”为例：

```
get_line(cmd); //采集按键数字
if(strncmp("1", cmd, 1)==0) { //如果为“1”
    gprs_call(cmd, strlen(cmd)); //调用“拨打电话”实现函数
}
```

- 8) 编码实现拨打电话模块功能，在 gprs.c 文件中 gprs_call 函数处：

```
void gprs_call(char *number, int num)
{
    tty_writecmd("at", strlen("at"));
    //发送拨打命令 ATD
    tty_write("atd", strlen("atd"));
    //发送电话号码
```

```

    tty_write(number, num);
    //发送分号和命令结束标志
    tty_write(";\\r", strlen(";\\r"));
    usleep(200000);
}

```

9) 分析函数 `tty_write()` 的作用;

10) 执行 `make` 命令编译该程序, 生成 `gprs` 可执行程序;

11) 利用 `mount` 命令挂载到开发板上, 然后运行 `gprs` 程序; 进行操作, 看是否能够实现拨打电话的功能。

3.4.2 补充发送短信模块的代码, 并编译、运行。

1) 在 `gprs.c` 文件中, 在 “//完善代码: 实现发送短信 `gprs_msg()` 函数” 注释行下面添加如下代码。实现短信息的发送功能。

```

void gprs_msg(char *number, int num)
{
    char ctl[]={26,0};           //ctl[0]=26,ctrl+z
    char text[]="gprs sms demo!";

    tty_writcmd("at", strlen("at"));
    //发送修改字符集命令:1,text 格式
    tty_writcmd("at+cmgf=1", strlen("at+cmgf=1"));
    //发送短信命令: 命令+号码+信息+结束符
    tty_write("at+cmgs=", strlen("at+cmgs="));
    //发送电话号码
    tty_writcmd(number, strlen(number));
    //发送文本信息
    tty_write(text, strlen(text));
    //发送结束符
    tty_write(ctl, 1);

    usleep(300000);
}

```

2) 编译、执行, 观察结果;

3) 请修改程序, 将个人的 “学号、姓名” 以短信形式发送出去, 看其他手机接收是否正常查看?

3.4.3 编码实现接听电话模块的功能。

在 `gprs.c` 文件中, 在 “//完善代码: 实现接听电话 `gprs_ans()` 函数” 注释行下面添加如下代码。实现电话的接听功能。

```

void gprs_ans()

```

```

{
    tty_writcmd("at", strlen("at"));
    //发送接听命令 ATA
    tty_writcmd("ata", strlen("ata"));
}

```

3.4.4 编码实现挂断电话模块的功能。

在 gprs.c 文件中，在“//完善代码：实现挂断电话 gprs_hold()函数”注释行下面添加如下代码。实现电话的挂断功能。

```

void gprs_hold()
{
    tty_writcmd("at", strlen("at"));
    //发送挂机命令 ATH
    tty_writcmd("ath", strlen("ath"));
}

```

编译项目，并运行调试；观察结果是否正确？

3.4.5 执行 gprs 程序，体验 gprs 程序的功能。

- 1) 拷贝本次练习编译后的程序 gprs 至 /arm2410/emlab 目录中。
- 2) 连接好实验开发板和开发宿主机（不要启动实验开发板）。
- 3) 连接好 gprs 模块、SIM 卡、天线，然后小心并正确插入实验开发板的扩展槽中。
- 4) 启动 minicom，给开发板加电，以 nfs 方法挂载 gprs 项目所在的目录。
- 5) 执行 gprs 程序，此时应能看到 gprs 初始化过程中的提示，然后显示一个命令菜单。
- 6) 在 keyshell 提示符下，利用小键盘输入“1”后回车，即开始拨打电话。然后会要求你输入所要拨打电话的号码，输入之后回车，观察结果是否能正常拨打电话？
- 7) 试一下发送短信的功能，看其他手机能否收到短信？